

Лабораторна робота №1

ІІЗ-21-5 Богайчук Денис

Мета: Дослідити математичну модель нейрона.

Хід роботи

Завдання 1. Реалізувати обчислювальний алгоритм для $\text{or}(x_1, x_2)$ і $\text{and}(x_1, x_2)$ в програмному середовищі (C++, Python, та ін.). Для реалізації обчислювальних алгоритмів рекомендується використання онлайн середовищ тестування (наприклад repl.it, [trinket](https://trinket.io), і т.д.).

Лістинг коду програми

```
import numpy as np
import matplotlib.pyplot as plt

# Функція активації
def step_function(x):
    return 1 if x >= 0 else 0

# Персептрон для функції OR
def or_perceptron(x1, x2):
    weights = np.array([1, 1]) # Ваги
    threshold = -0.5 # Попіг
    inputs = np.array([x1, x2]) # Вхідні дані
    linear_combination = np.dot(weights, inputs) + threshold # Лінійна комбінація
    return step_function(linear_combination)

# Персептрон для функції AND
def and_perceptron(x1, x2):
    weights = np.array([1, 1]) # Ваги
    threshold = -1.5 # Попіг
    inputs = np.array([x1, x2]) # Вхідні дані
    linear_combination = np.dot(weights, inputs) + threshold # Лінійна комбінація
    return step_function(linear_combination)

# Генерація випадкових точок
np.random.seed(42) # Для відтворюваності результатів
num_points = 200 # Кількість точок
x_random = np.random.rand(num_points) * 2 - 0.5 # Генерація значень від -0.5 до 1.5
y_random = np.random.rand(num_points) * 2 - 0.5 # Генерація значень від -0.5 до 1.5

# Класифікація точок за допомогою персептронів OR та AND
or_results = np.array([or_perceptron(x, y) for x, y in zip(x_random, y_random)])
and_results = np.array([and_perceptron(x, y) for x, y in zip(x_random, y_random)])
```

```

# Встановлення кольорів: синій для 1, оранжевий для 0
colors_or = ['blue' if result == 1 else 'orange' for result in or_results]
colors_and = ['blue' if result == 1 else 'orange' for result in and_results]

# Створення графіків
fig, ax = plt.subplots(1, 2, figsize=(12, 6))

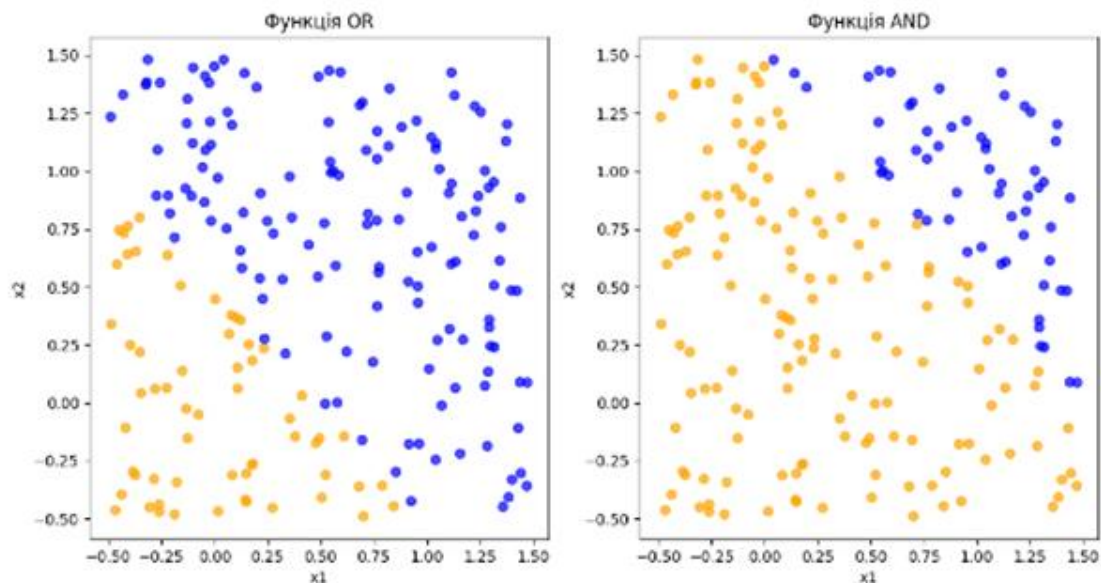
# Графік для OR
ax[0].scatter(x_random, y_random, c=colors_or, alpha=0.7)
ax[0].set_title('Функція OR')
ax[0].set_xlabel('x1')
ax[0].set_ylabel('x2')

# Графік для AND
ax[1].scatter(x_random, y_random, c=colors_and, alpha=0.7)
ax[1].set_title('Функція AND')
ax[1].set_xlabel('x1')
ax[1].set_ylabel('x2')

plt.show()

```

Результат виконання коду програми



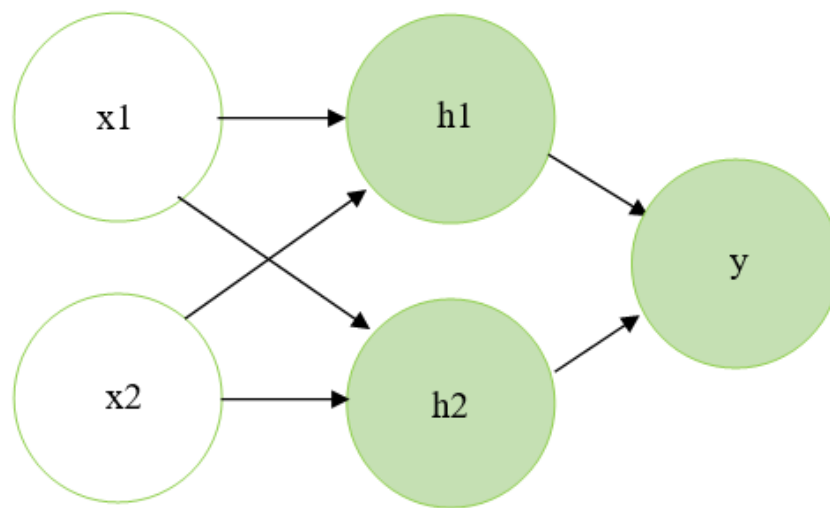
Код генерує випадкові числа в діапазоні від 0 до 1, які потім масштабуються до інтервалу від -0.5 до 1.5. Для забезпечення повторюваності результатів використовується фіксоване початкове значення генератора. У цьому коді аналізується робота персептрона на логічних функціях OR та AND: задаються значення ваг і порога, розраховується зважена сума вхідних даних з урахуванням порога, а результат визначається за допомогою функції активації. Усі результати представлені у вигляді

візуалізації.

Завдання 2. Зобразити двохслойний персептрон для функції $\text{xor}(x_1, x_2)$ та скласти відповідне рівняння розділяючої прямої, використовуючи теоретичний матеріал даної лабораторної роботи.

$$\text{XOR}(x_1, x_2) = \text{OR}(\text{AND}(x_1, \text{NOT}(x_2)), (\text{AND}(\text{NOT}(x_1), x_2)))$$

Персептрон функції XOR.



Функція XOR не може бути розділена на два класи за допомогою однієї прямої, як це можливо для функцій AND або OR, оскільки вона не є лінійно роздільною. Однак двошарова нейронна мережа може створити дві окремі прямі, які дозволяють виділити області, відповідні істинним і хибним значенням XOR.

Розглянемо першу лінію для h_1 : $h_1 = \sigma((-1) * x_1 + x_2 + 1)$

Лінійне рівняння для цієї межі можна виразити як: $-x_1 + x_2 + 1 = 0$ або $x_2 = x_1 - 1$.

Визначаємо другу пряму для h_2 : $h_2 = \sigma(x_1 + (-1) * x_2 + 1)$

Лінійне рівняння цієї межі можна записати: $x_1 - x_2 + 1 = 0$ або $x_2 = x_1 + 1$.

Висновок: У процесі виконання завдань підтверджено, що функції OR та AND можуть бути обчислені за допомогою однорівневого персептрона, тоді як функція XOR вимагає використання двошарової нейронної мережі. Отримані результати демонструють ефективність та логічну коректність підходу, застосованого до аналізу цих логічних функцій.